# Secured Artificial Intelligence Based Face Anti-spoofing Detection Model via Serverless Architecture and SaaS Based Cloud Platform

Sai Sanjay Kottakota * and Anantha Gnaneswar *

Posted Date: 10 September 2024

doi: 10.20944/preprints202409.0740.v1

Keywords: Serverless Architecture; Facial Recognition Security; Spoof Attack Prevention; Cloud Computin; Cybersecurity; Deep Learning; Threat Mitigation

 doi:10.20944/preprints202409.0740.v1



*Article*

# Secured Artificial Intelligence Based Face Anti-Spoofing Detection Model via Serverless Architecture and SaaS Based Cloud Platform

**Sai Sanjay Kottakota * and Anantha Gnaneswar**

School of Computer Science and Engineering, VIT-AP UNIVERSITY; gnanesh.anthh@gmail.com
* Correspondence: saisanjay7660@gmail.com

**Abstract:** As the adoption of deep learning models continues to surge across various applications, the need for efficient deployment architectures becomes increasingly critical. This paper presents a novel approach to enhance the deployment of deep learning models by leveraging serverless architecture. Serverless computing has been popular for its auto-scaling, cost-effectiveness, and simplified management characteristics. However, the intense resource demands of deep learning models pose challenges in maintaining low response times and effective load balancing within serverless environments. The proposed architecture addresses these challenges by integrating principles from both deep learning model optimization and serverless computing. Through systematic experimentation and analysis, we demonstrate that by appropriately designing and tuning the deployment architecture, significant improvements in response time, performance, resource utilization, and load distribution can be achieved.

**Keywords:** serverless architecture; facial recognition security; spoof attack prevention; Cloud computin; cybersecurity; deep learning; threat mitigation

## 1. Introduction

In today's digital age, the reliance on facial recognition technology has become increasingly prevalent across many applications, from unlocking our smartphones to enhancing security in critical infrastructure and public spaces. However, as facial recognition systems have gained prominence, so too have concerns regarding their vulnerability to spoof attacks. In a world where identity theft and data breaches emerge as significant threats, developing robust face anti-spoofing detection models is really important.

Face anti-spoofing detection models serve as the first line of defense against malicious actors seeking to deceive facial recognition systems through various means, such as presenting photographs, masks, or even 3D-printed replicas. These attacks not only compromise security but also raise privacy and ethical concerns. As such, research and advancements in face anti-spoofing detection models have gained considerable attention from the scientific community, security experts, and policymakers. Developing a facial anti-spoofing system with improved recognition performance, faster response times, and greater resilience is crucial.

These types of fraud detection systems that use deep learning models can be developed using Azure Functions, which is a serverless concept of cloud computing that allows a piece of code to be deployed and executed without needing server infrastructure, web server, or any configurations. Serverless architecture represents a groundbreaking approach to cloud computing, offering an exceptionally efficient and scalable solution for deploying deep learning models. This innovative paradigm eliminates the need for managing servers, allowing developers to concentrate exclusively on their code and applications. Serverless computing delivers a multitude of advantages that make it an ideal choice for deep learning deployment. First and foremost, it excels in cost-efficiency, as organizations pay only for the computing resources they use, thereby reducing operational expenses. Furthermore, serverless platforms provide a remarkable degree of scalability, automatically adapting to varying workloads and large datasets, all without the necessity for manual intervention. Moreover, serverless architecture

 doi:10.20944/preprints202409.0740.v1



simplifies management by removing the burden of server maintenance and provisioning. It also ensures low-latency responses, vital for real-time deep learning applications. Additionally, it offers developers the flexibility to work with their preferred deep learning frameworks and tools. Serverless architecture's parallel processing capabilities speed up model training and inference, and its automatic scaling handles unpredictable workloads. This architecture's fault tolerance, easy integration with other services, and rapid deployment capabilities further enhance its appeal as an invaluable tool for deploying deep learning models in an agile, cost-effective, and scalable manner.

## 2. Literature Survey

[1] Serverless computing is an emerging cloud paradigm that provides transparent resource management and scaling for users, making it attractive for ML design and training developers. Function-as-a-Service (FaaS) and Container-as-a-Service (CaaS) are widely-realized forms of serverless computing that offer fine-grained resource management and flexible billing models. They have been studied for ML model serving and training, showcasing benefits such as scalability and cost efficiency. ML workflows with continuous learning and training can benefit from dynamic resource allocation for performance and cost optimizations. However, the communication overhead and resource limitations in serverless platforms can affect the feasibility of training large-scale models. Hybrid storage solutions, combining fast storage mediums like in-memory key-value stores with cloud-based object stores, have been found to scale well for large ML models. They can satisfy latency-sensitive demands and strike a balance between performance and cost.

[2] This paper introduced an operational classification and a four-layered structure for deploying digital health models in the cloud. The four layers encompass containerized microservices for ease of maintenance, a serverless architecture for enhanced scalability, function as a service for enhanced portability, and FHIR schema for improved discoverability. This customized architecture proves highly effective for applications intended for use by downstream systems like EMRs and visualization tools. They presented a taxonomy centered around workflows to support the practical implementation of this approach. Recognizing FHIR as a burgeoning standard for healthcare interoperability, the proposal suggests employing FHIR schema for seamlessly integrating ML application programming interfaces (APIs) into existing health information systems.

[3] The presented work addresses the challenges of deploying deep neural network models in real-time, emphasizing the contrast between the time-consuming training phase and the stringent throughput and latency requirements during model inference. While high-performance clusters are traditionally used for inference, their maintenance cost can be prohibitive. The paper introduces serverless computing as a cost-effective alternative, where the pricing model is based on execution time, abstracting away infrastructure management complexities. The serverless approach is discussed in the context of deploying machine learning and deep learning applications, focusing on its benefits such as cost-effectiveness, ease of scalability, and abstraction of infrastructure management. However, the authors highlight that serverless architecture might not be universally applicable, necessitating developers to optimize its use based on specific application requirements.

The work proposes a methodology for migrating vision algorithm-based applications, particularly those containing a suite of models, from on-premise deployment to a serverless architecture. The study evaluates the cost and performance of serverless architecture compared to on-premise deployment and virtual machine instances on the cloud. Additionally, the authors explore the impact of using multiple cloud services on the performance of Function-as-a-Service (FaaS) for implementing large models.The paper presents optimizations to overcome serverless architecture constraints, including trimming TensorFlow framework, loading input data efficiently, and leveraging Elastic File System (EFS) for storing large models. The experimental results demonstrate the performance and cost implications of serverless computing, comparing it with on-premise and virtual machine deployments. The study includes an in-depth analysis of factors such as response time, throughput, cold start effects, memory size influence, and cost considerations.

 doi:10.20944/preprints202409.0740.v1



[4] The paper proposes a novel architecture for serving deep learning models through APIs via a SaaS platform. The architecture aims to provide a scalable and efficient solution for deploying and serving deep learning models in a cloud-based environment. The authors highlight the importance of APIs in enabling easy access and integration of deep learning models into various applications. The proposed architecture leverages the benefits of a SaaS platform to provide a seamless and user-friendly experience for developers and users. The authors discuss the challenges and considerations in designing such an architecture, including scalability, security, and performance. The paper presents a detailed technical overview of the architecture, including the components and their functionalities. The authors also provide experimental results to demonstrate the effectiveness and efficiency of the proposed architecture.

[5] This paper explores the serverless paradigm and introduces the notion of Function as a Service (FaaS) as an innovative framework for developing applications and services. It introduces the Apache OpenWhisk, a distributed serverless platform driven by events, as a means to implement Machine Learning Functions as a Service (ML-FaaS) and construct pipelines for machine learning applications. The proposed approach leverages the Apache OpenWhisk serverless platform to create a custom-made chain of functions for building serverless applications. These functions address specific machine learning tasks, including data pre-processing and training ML classifiers. The paper presents a two-phase hybrid ML-FaaS approach, consisting of an offline phase and an online phase. The offline phase involves building a pipeline of functions in a serverless ecosystem, while the online phase handles the processing of new data through the pipeline. The paper highlights the need for new frameworks and functionalities in serverless environments to optimize resource management, scalability, parallelism, cost-effectiveness, and latency issues. It emphasizes the potential of serverless models in enabling flexible extensions and dynamic workflows for analytics tasks. The assessment results of the suggested methodology encompass the response time of the executed pipeline upon the initiation of a request. The approach showcases its efficiency in handling data and delivering outcomes within a matter of milliseconds.

[6] This paper discusses the evolution of cloud computing over the last decade, emphasizing its impact on virtualized computing and the emergence of service delivery models such as IaaS, PaaS, and FaaS. The focus shifts to FaaS, exemplified by AWS Lambda and Azure Functions, highlighting its event-driven execution capabilities and advantages over traditional IaaS offerings. The integration of lightweight virtualization technologies, including Linux containers, Docker, and Container Orchestration Platforms like Kubernetes, paved the way for serverless computing. The paper addresses the limitations of current public cloud serverless offerings for scientific computing, leading to the development of an open-source platform supporting hybrid data processing workflows across on-premises and public cloud environments. A smart city use case involving video surveillance and face mask detection illustrates the platform's efficiency, with experiments demonstrating the cost-effectiveness of offloading computing-intensive tasks to AWS Lambda. The survey concludes with insights into future work, emphasizing dynamic resource orchestration across the Cloud-to-Things continuum and the adaptation of serverless computing for edge devices and IoT.

[7] This paper discusses the critical shift from traditional client-server architectures to serverless architectures, particularly in the deployment of AI workloads. Standard architectures are shown to face scalability issues, reliability compromises, and increased complexity, leading to a growing trend towards serverless or microservices-based solutions. Serverless architectures, exemplified by platforms like AWS Lambda, offer advantages such as automatic scalability, simplified development pipelines, and cost savings. However, they come with constraints, especially for AI workloads, including limited deployment package size and absence of GPU support. The paper proposes a suite of optimization techniques addressing these constraints, encompassing the minimization of Python libraries, dynamic loading of AI models into temporary runtime memory, a two-step ML process with ONNX formatted models, and innovative data handling techniques. Evaluation of these techniques, using examples from the Real-Time Flow project, demonstrates their effectiveness in transforming complex AI workloads

 doi:10.20944/preprints202409.0740.v1



for serverless deployment, particularly in predicting train delays based on large datasets. The study emphasizes the importance of overcoming limitations in deployment environments for real-time AI applications, providing a comprehensive overview of the proposed optimization strategies.

## 3. Proposed Solution

In our proposed solution, the user initiates the process by sending an image from a Biometric Device for spoof or real testing. This image is transmitted to the API endpoint via a secure POST request, and it is securely delivered to the Azure Functions endpoint with base64 encoding. Within the Azure Functions, the image undergoes a series of meticulously orchestrated steps for comprehensive facial anti-spoofing assessment.

The base64-encoded text is initially decoded, transforming it into a numpy array. This decoded image is then subjected to an intricate process facilitated by the powerful OpenCV library, operating in conjunction with the cutting-edge Mediapipe framework. This collaborative effort allows us to extract predefined Facial Landmark Coordinates, enabling us to precisely locate the face within the image.

Finally, the real crux of our system comes to the forefront as the facial anti-spoofing model is engaged for inference. This model is powered by PyTorch and adeptly runs on the CUDA framework. During this inference step, the model delves deep into the image, effectively determining whether the captured image portrays a genuine or fake face. The essence of our solution lies in the intelligent amalgamation of these sophisticated technologies, enabling our system to deliver precise and robust anti-spoofing assessments.

### *3.1. Model Training and Data Enhancement*

Our approach hinges on the use of the ResNet-18 deep learning model, celebrated for its prowess in feature extraction. However, it's important to note that the choice of this model is far from arbitrary. Instead, it's a result of careful selection, followed by rigorous training on a diverse and extensive dataset. This training is an iterative process where the model learns to differentiate between authentic facial images and their deceitful, spoofed counterparts. To achieve this level of discrimination, custom loss functions such as Similarity Loss and AdMSoftmax Loss are diligently employed to fine-tune the model's performance. These loss functions play a pivotal role in equipping the model with the capability to discern the subtle nuances that distinguish real faces from fraudulent or synthetic ones. The system is thus primed to tackle a broad spectrum of spoofing attempts, ensuring the security and reliability of facial recognition processes.

### *3.2. Facial Detection and Preprocessing*

At the heart of our methodology lies the inclusion of MediaPipe Face Detection, a state-of-the-art framework recognized for its proficiency in facial detection and tracking. This integral component is instrumental in accurately identifying the face within the provided image. What sets it apart is its adaptive bounding box adjustment, which seamlessly encompasses essential contextual information around the detected face. This adaptive approach significantly enhances the overall accuracy of anti-spoofing assessments, making our system resilient to various presentation styles, occlusions, and spoofing attempts. Once the face is precisely located, a sequence of preprocessing steps commences to prepare the detected face for deep learning analysis. An essential step involves resizing the face to a standardized 256x256 resolution. This preprocessing not only ensures uniformity in the model's input but also greatly optimizes subsequent inference processes, enabling the system to operate with precision and speed.

### *3.3. Inference and Classification*

The pinnacle of our system is reached in the inference stage, where the facial anti-spoofing model takes center stage. This model, running on PyTorch and utilizing the power of the CUDA framework, becomes the decision-maker, effectively discerning whether the captured image authentically portrays

 doi:10.20944/preprints202409.0740.v1



a real face or if it is a deceptive representation. Throughout this process, the model's deep neural network architecture meticulously scrutinizes the image, extracting and analyzing intricate features and patterns. It then confidently labels the image as 'Real' or 'Spoof,' providing confidence scores that further validate the authenticity of the determination. Our system hinges on the synergy of these meticulous processes, providing users with a highly secure and dependable facial anti-spoofing detection service.

### *3.4. System Architecture*

Our system architecture is a harmonious blend of cutting-edge technology and scalability. The core components of the architecture include:

Azure Functions: Our face anti-spoofing detection model finds its home within Azure Functions, a serverless computing service. This architectural choice offers multifaceted advantages. It ensures on-demand scalability, enabling the system to seamlessly adapt to varying workloads. Automatic resource management is another hallmark feature, eliminating the need for manual intervention in resource allocation. Furthermore, Azure Functions align with efficient cost management practices, guaranteeing economical utilization of cloud resources.

Deep Learning Model: The ResNet-18-based deep learning model is meticulously integrated into the Azure Function. Whether running on a CPU or a GPU, this model is primed for efficient inference, enabling swift and accurate anti-spoofing assessments.

MediaPipe Integration: Our system's face detection prowess is fortified with the integration of the MediaPipe framework. Renowned for its real-time cross-platform capabilities in facial landmark detection and tracking, MediaPipe plays a pivotal role in ensuring the precise and comprehensive detection of faces.

Azure Content Delivery Network (CDN) :In our application, we utilize Azure CDN to enhance the distribution of our web-based face anti-spoofing detection service. Azure CDN is a global network of strategically placed data centers designed to accelerate content delivery to users. By caching static assets, such as JavaScript files, cascading style sheets, and images, at edge locations, we ensure faster content delivery to users regardless of their geographical location. This not only optimizes the responsiveness of our web service but also enhances its scalability, making it suitable for a global audience while maintaining low-latency interactions.

Azure Front Door: In our application, Azure Front Door is integrated to ensure robust and highly available access to our face anti-spoofing detection service. It serves as a traffic manager, directing users to the nearest Azure datacenter for low-latency interactions. By intelligently routing requests and providing redundancy, Azure Front Door guarantees the reliability and performance of our service, even in the face of high demand or unexpected traffic spikes. Moreover, it enhances security through Web Application Firewall (WAF) and DDoS protection, safeguarding our application from potential threats and ensuring uninterrupted service availability.

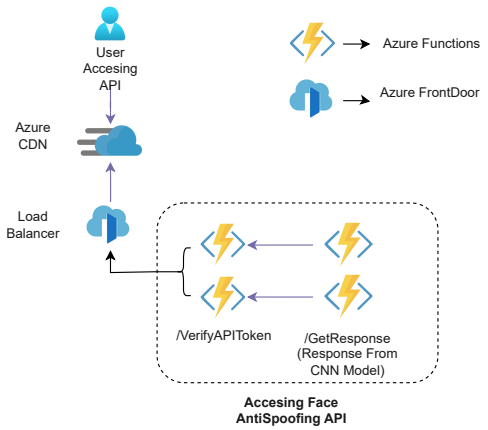


**Figure 1.** Architecture.

 doi:10.20944/preprints202409.0740.v1



### *3.5. Web Deployment*

Our solution's accessibility is streamlined through a web-based API. Users can conveniently submit images for anti-spoofing assessments by embedding the image within the request body. The image is then converted into base-64 encoding format and sent to the azure function for further processing. Finally, the user gets a reply on the screen, providing users with a clear and structured response. This response encapsulates the predicted label ('Real' or 'Spoof') and confidence scores, making the output easily interpretable and actionable.

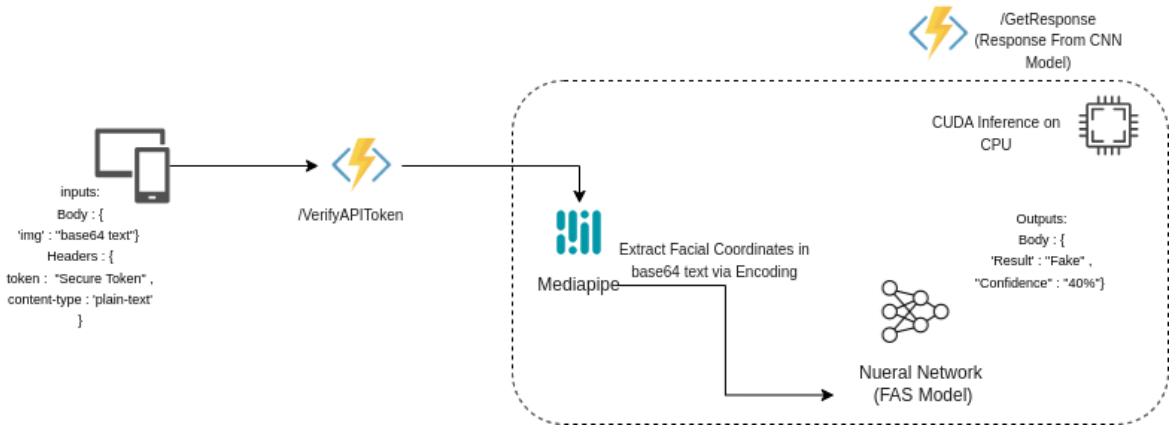


**Figure 2.** System Data Flow.

## 4. Results and Discussion

Our face anti-spoofing detection system underwent rigorous evaluation to ascertain its effectiveness and reliability in distinguishing real facial images from spoofed ones. The system demonstrated a remarkable accuracy rate in correctly identifying genuine faces and discerning them from spoofed attempts. Our model consistently achieved a remarkable accuracy, reaffirming its robustness in making accurate anti-spoofing determinations.

The system demonstrated robustness against a diverse set of spoofing attempts, including printed photos and facial masks. The dynamic bounding box adjustment in face detection played a pivotal role in ensuring robustness against various spoofing strategies.

### *4.1. Compiling and processing of Datasets*

we used three various unique databases that contain fake and spoofed images to test with the model so that it will be more robust and secure and work with various scenarios and is not limited a single feature detection.

ROSE : youtu face liveness detection dataset This dataset contains a wide array of lighting situations, various sorts of attacks and various camera models, making it an effective asset for training anti-spoofing models that can handle different scenarios. The dataset contains 55,000 counterfeit photo samples. It provides a diverse range of fake face images, offering a rich variety for analysis and model development.

NUAA Photograph Imposter Database: This dataset provides a more balanced set with legit and fake pictures. It contains 7,509 spoofed faces and 5,106 authenitc faces making a total of 12,615 images.

Large Crowd collected Face Anti-Spoofing Dataset: This one has 1,943 legitimate faces and 16,885 spoofed faces. It is more diverse and real life based because of the amount of various backgrounds and situations of the photos taken.

OULU-NPU Face Presentation Attack Database: The inclusion of the OULU-NPU Face Presentation Attack Database significantly enhances the test dataset. It encompasses 4,950 videos depicting genuine access instances and attacks, recorded using front cameras from six mobile devices. These videos are further categorized into developing, training and inspecting sub datasets.

 doi:10.20944/preprints202409.0740.v1



Replay-Attack Database: The Replay-Attack Database dedicated to face spoofing comprises 1,300 video clips portraying attempts at photo and video-based attacks on 50 distinct clients. These clips vary across diverse lighting conditions and are created in two primary ways: firstly, by genuine clients attempting to enter a laptop using the webcamera, and secondly, by displaying a video recording of the same client for a minimum duration of 9 seconds.

MSU-MFSD Dataset: The MSU-MFSD Dataset refers to the Mobile-Simulated Unconstrained Multispectral Face Spoof Detection Dataset. It contains a comprehensive collection of facial images captured in various spectral bands, including the visible and near-infrared spectrum. This dataset is specifically designed for developing and evaluating face anti-spoofing algorithms, aiming to enhance the accuracy and robustness of facial recognition systems against spoof attacks in diverse environmental conditions and illumination settings.

You can see the classification between live and spoof images for each dataset in Figure 3. it tells you how many classes each category of a dataset has.

| Dataset | Classes | |
|---|---|---|
| | live | spoof |
| OULU-NPU | 1 | 4 |
| MSU-MFSD | 2 | 6 |
| REPLAY ATTACK | 1 | 3 |
| ROSE-YOUTU | 1 | 3 |
| LCC-FASD | 1 | 1 |
| NUAA | 1 | 1 |

**Figure 3.** Comparison of Datasets.

The deployment of our model within a serverless architecture was instrumental in ensuring rapid inference. The response times were consistently within acceptable limits for real-time applications, thereby meeting the demands of time-sensitive scenarios.

The results of our face anti-spoofing detection system signify a significant step forward in enhancing the security of facial recognition processes. The system's exceptional accuracy and low latency make it well-suited for real-world applications where security is paramount. One of the key strengths of our solution lies in the integration of MediaPipe Face Detection, which not only locates the face but also intelligently adapts the bounding box to encompass additional contextual information. This adaptive approach mitigates potential challenges posed by variations in presentation, facial occlusions, or attempts at spoofing. Moreover, the serverless architecture ensures scalability, cost-efficiency, and rapid response times, making our system suitable for deployment in a wide array of scenarios. However, it is important to acknowledge that no system is without its limitations. While our solution excels in many aspects, there are still areas for improvement. Notably, further research and development are required to enhance its resilience against advanced spoofing techniques and to expand its adaptability to different cultural and demographic groups. Moreover, continuous updates and model retraining are essential to address emerging threats in spoofing attempts. Additionally, ensuring the privacy and ethical use of facial recognition technology is an ongoing concern, and our system is designed to comply with privacy regulations and ethical considerations.

 doi:10.20944/preprints202409.0740.v1



## 5. Conclusions and Future Scope

In this research, we have presented a robust and efficient solution for facial anti-spoofing detection. By integrating cutting-edge deep learning models within a serverless architecture and a Software as a Service (SaaS) platform, we have successfully addressed the critical challenge of distinguishing genuine facial images from fraudulent or spoofed attempts. Our approach leverages a carefully trained ResNet-18 model, coupled with custom loss functions, to achieve a high degree of accuracy in anti-spoofing assessments. The inclusion of MediaPipe Face Detection further enhances our system's robustness, ensuring precise face location and adaptability to diverse presentation styles. The model's inference phase, running on the CUDA framework, solidifies our system's capability to make reliable determinations regarding the authenticity of captured facial images. This research marks a significant advancement in facial recognition security, offering a versatile solution applicable across various domains, from access control to identity verification.

While our research has delivered a robust anti-spoofing solution, there are promising avenues for future exploration and enhancement. Firstly, the rapid evolution of spoofing techniques necessitates ongoing research to improve system resilience against novel threats, including deepfake technology and 3D masks. Secondly, extending the system's adaptability to a diverse range of demographic groups is crucial, ensuring equitable and accurate assessments for users from different backgrounds. Additionally, as facial recognition technology raises ethical and privacy concerns, future work must continue to address these issues through stringent compliance with regulations and ethical guidelines. Further refinement of the user interface and experience is essential to ensure user-friendliness and accessibility for a broader user base. The integration of real-time monitoring and alerting systems can provide immediate notifications in case of spoofing attempts, enhancing security. Lastly, regular model updates and retraining are vital to keep the system up-to-date with emerging spoofing methods and to maintain high accuracy levels. In conclusion, our research serves as a foundation for a secure and reliable facial anti-spoofing detection system, and the future holds exciting prospects for improving this technology to ensure its adaptability to evolving challenges and its adherence to ethical standards, ultimately making it a cornerstone in the realm of biometric security and identity verification.

## References

1. Ali, Ahsan, et al. "SMLT: A Serverless Framework for Scalable and Adaptive Machine Learning Design and Training." arXiv, 4 May. 2022, doi:10.48550/arXiv.2205.01853.
2. Eapen, Bell Raj, et al. "Serverless on FHIR: Deploying machine learning models for healthcare on the cloud." arXiv, 8 June 2020, doi:10.48550/arXiv.2006.04748.
3. Dheeraj, Chahal., Ravi, Ojha., Manju, Ramesh., Rekha, Singhal. (2020). Migrating Large Deep Learning Models to Serverless Architecture. 111-116. doi: 10.1109/ISSREW51248.2020.00047
4. A. Christidis, R. Davies and S. Moschoyiannis, "Serving Machine Learning Workloads in Resource Constrained Environments: a Serverless Deployment Example," 2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA), Kaohsiung, Taiwan, 2019, pp. 55-63, doi: 10.1109/SOCA.2019.00016.
5. E. Paraskevoulakou and D. Kyriazis, "Leveraging the serverless paradigm for realizing machine learning pipelines across the edge-cloud continuum," 2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), Paris, France, 2021, pp. 110-117, doi: 10.1109/ICIN51074.2021.9385525.
6. Risco, S., Moltó, G., Naranjo, D.M. et al. Serverless Workflows for Containerised Applications in the Cloud Continuum. J Grid Computing 19, 30 (2021). https://doi.org/10.1007/s10723-021-09570-2
7. A. Christidis, S. Moschoyiannis, C. -H. Hsu and R. Davies, "Enabling Serverless Deployment of Large-Scale AI Workloads," in IEEE Access, vol. 8, pp. 70150-70161, 2020, doi: 10.1109/ACCESS.2020.2985282.